



AF
LFW

PATENT APPLICATION

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of

Docket No: A8642 / ST9-99-151

SHEN, HongHai

Appln. No.: 09/512,738

Group Art Unit: 2178

Confirmation No.: 5283

Examiner: Joshua D. Campbell

Filed: February 24, 2000

For: PROVIDING DYNAMIC WEB PAGES BY SEPARATING SCRIPTS AND HTML
CODE

REPLY BRIEF PURSUANT TO 37 C.F.R. § 41.41

MAIL STOP APPEAL BRIEF - PATENTS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 41.41, Appellant submits this *Reply Brief* in response to the *Examiner's Answer* dated October 21, 2005. Entry of this *Reply Brief* is respectfully requested.

Table of Contents

PROCEDURAL SECTIONS.....	2
REPLY TO EXAMINER'S "RESPONSE TO ARGUMENT"	3
CONCLUSION.....	18

Reply Brief Under 37 C.F.R. § 41.41
U.S. Appln. No.: 09/512,738

Attorney Docket # A8642 /
ST9-99-151

PROCEDURAL SECTIONS

Appellant concurs as to the accuracy of the procedural statements in sections 1-9 of the
Examiner's Answer.

REPLY TO EXAMINER'S "RESPONSE TO ARGUMENT"

Appellant continues to disagree with the substance of the current rejections. In the sections below, Appellant explains in detail these continued disagreements.

For the Board's convenience, Appellant notes that each of the sections A(1) - A(12) below has a direct correspondence to section letters C(1) - C(12) in the August 5, 2005 *Appeal Brief*.

A(1). Lipkin Fails To Teach Or Suggest Transforming "The First Object By Changing The Value Thereof In Accordance With The Transformation Instruction," as Recited in Claims 1, 11 and 21

Appellant argued in the August 5, 2005 *Appeal Brief* that the proffered combination of *Jamtgaard* and *Lipkin* fails to teach or suggest independent claim 1, 11 and 21's recitation of transforming "the first object by changing the value thereof in accordance with the transformation instruction."

More specifically, Appellant pointed out that *Lipkin* discloses only:

(1) the insertion of new command lines containing processing instructions (col. 57, lines 7-40) and new links (col. 58, lines 19-58) to create the contents of the viewable page that is to be displayed, *i.e.*, the insertion of new nodes rather than the claimed changing of the values of existing nodes; and

(2) the formatting of the XSP model page for a particular display type by applying "widgets" and an XSLT stylesheet thereto, *i.e.*, the changing of the format of a model page rather than the claimed changing of the values of existing nodes.

While the Examiner seems to concede that Appellant's point (2) above is correct (as he provides no counter-argument), the Examiner seems to disagree with Appellant's point (1) above, and alleges that (see *Answer*, p. 11, lines 12-16 and p. 12, lines 10-15):

[t]he Lipkin reference teaches that a transformation instruction can be used to change the value of an object in a DOM tree in a way that is transparent to the user (column 69, lines 1-54 of Lipkin). As it is shown in Lipkin any instruction to transform or manipulate the DOM tree is directed through the first object (root node) because that is how all other nodes/objects are accessed (column 69, lines 1-24 of Lipkin) ... [; and] ...

[i]n column 69, lines 1-54, Lipkin shows that during a transformation process a DOM tree is accessed and the values of the nodes are changed. In this case, the sub-tree be [sic] inserted into the DOM tree is copied to the nodes that require it, thus changing the valued [sic] of those nodes specifically...

Appellant respectfully disagrees with the Examiner's assertion that *Lipkin* "teaches that a transformation instruction can be used to change the value of an object in a DOM tree." Rather, the section of *Lipkin* cited by the Examiner (col. 69, lines 1-54 of *Lipkin*) specifically indicates that a "DOM node will be inserted in the Document" (col. 69, lines 19-20). Accordingly, Appellant respectfully submits that *Lipkin* discloses inserting new command lines containing processing instructions (col. 57, lines 7-40) and new links (col. 58, lines 19-58) to create the contents of a viewable page for display. There is no teaching or suggestion that the insertion of such command lines and links changes the value of the underlying objects.

As a matter of example, Appellant directs the Board to col. 67, lines 1-25 of *Lipkin*, which shows a model document (*i.e.*, before insertion of command lines) and to col. 68, lines 1-23 of *Lipkin*, which shows the model document post-processing (*i.e.*, after insertion of command lines). Clearly, when comparing these two documents, it is apparent that the any feature that

could be alleged to be an “object: with a “value” does not change in the post-processing document (in col. 68, lines 1-23).

In sum, Appellant respectfully submits that *Lipkin*’s disclosed process of inserting new command lines is simply not equivalent to the recited transformation of “the first object by changing the value thereof in accordance with the transformation instruction.”

A(2). One Of Skill Would Not Have Been Motivated to Modify Jamtgaard in View of Lipkin To Arrive At The Features of Independent Claims 1, 11 and 21

Appellant argued in the August 5, 2005 *Appeal Brief* that, even if *Lipkin* could be read as somehow disclosing an underlying modification of values of DOM objects (which Appellant does not concede), one of skill would still not have been motivated to modify *Jamtgaard* in view of *Lipkin*, as the Examiner alleges.

More specifically, Appellant pointed out that the Examiner’s proffered motivation to modify *Jamtgaard* in view of *Lipkin* - “to allow for a transformation to occur that is transparent to the user” - does not provide the motivation required for a *prima facie* showing of obviousness, because:

(1) *Jamtgaard* already provides an “automatic” reformatting of content that is transparent to the user;

(2) *Jamtgaard* is directed to reformatting an existing document so that it may be displayed on a particular client, but *Lipkin* is differently directed to creating and formatting a document from scratch so that it may be viewed on a particular client; and

(3) the *Lipkin* system must be used in conjunction with an XSP model page as its starting point, while *Jamtgaard* reformats an existing mark-up language page.

The Examiner seems to disagree with Appellant's point (1) above, and argues that (see

Answer, p. 11, line 16 - p. 12, line 3):

The transformation that occurs in *Jamtgaard* is a method of changing the format of the document, thus transforming the document into a format or output style that more readily agrees with a client device, while the transformation that occurs in *Lipkin* is a method of changing the contents/values of the DOM tree explicitly, thus changing the actual content by adding, editing, or deleting content. The combination of these two reference [sic] would have provided a method in which both the format/style of a document and the actual content of a document could be changed and adapted in a way that is transparent to the user, thus providing the user with a document specifically catered in content and format.

Appellant respectfully disagrees. While the Examiner alleges that provision of different "actual content" (such as is allegedly disclosed by *Lipkin*) would be desirable in *Jamtgaard*, there is no teaching or suggestion to support such a position. In contrast to such an arrangement, *Jamtgaard* is directed to providing the same underlying data in different formats to be viewable on different devices - *i.e.*, "a single piece of content that is re-formatted automatically for the different information appliances" (*Jamtgaard* Abstract). Thus, the Examiner's proffered modification of *Jamtgaard* to somehow also change the underlying data would be contrary to this stated purpose, as the alleged modification would result in a system that would no longer provide the same content in different formats, but would somehow provide different content in different formats.

Further, the Examiner seems to disagree with Appellant's point (2) above, and argues that (see *Answer*, p. 12, lines 15-18):

In the column 54, lines 36-47, *Lipkin* teaches that pages may be generated and preexisting pages can be customized and/or modified to provide different versions to user agents using the

cited embodiment of Lipkin, which is done automatically thus transparent to the user.

Appellant respectfully disagrees. *Lipkin* does not teach or suggest that its specific process can be applied to “preexisting pages,” as the Examiner alleges. Rather, the section cited by the Examiner (col. 54, lines 36-47) merely indicates that “Web Content Server 800 would allow developers to create web pages ... [making] it easier to customize pages, to provide different versions of pages to different user agents.” Thus, this section merely indicates that the final product (*i.e.*, viewable page) created by *Lipkin* may be customized (which, incidentally, is the entire purpose of *Lipkin*). There is no teaching or suggestion in this section, or any other portion of *Lipkin*, that *Lipkin's* system is directed to the modification of existing pages. Rather, as mentioned throughout the prosecution of this Application, *Lipkin* is directed to an express method of creating web pages using an XSP model page as its starting point (*e.g.*, see col. 63, lines 15-19; col. 66, lines 9-14).

Still further, the Examiner seems to disagree with Appellant's point (3) above, and argues that (see *Answer*, p. 13, lines 5-9):

... the teachings of Lipkin are directed towards creating and modifying HTML and XML documents (column 49, line 60 - column 50, line 7 of Lipkin), two of the most common types of mark-up language documents, using XSL which is also used in the teachings of Jamtgaard, thus the system would be operable with the mark-up language documents of Jamtgaard.

Appellant respectfully disagrees. While *Lipkin* is directed toward providing an HTML document as an end product, *Lipkin's* process necessarily utilizes an XSP model page as its beginning step (see, *inter alia*, col. 63, line 15-20; col. 64, line 60-65; col. 65, lines 30-35; col. 66, lines 10-15 of *Lipkin*). As discussed throughout the prosecution of this Application, the XSP

model page then has command lines inserted therein, and a style sheet applied thereto, to form a mark-up language page for viewing by a client.

In contrast, such an XSP model page is not even considered by *Jamtgaard*. Rather, *Jamtgaard* starts with an existing mark-up language page, and then reformats it to be viewable on different devices (*Jamtgaard* Abstract).

Thus, *Lipkin* and *Jamtgaard* cannot reasonably be considered as alternative or consecutive processes, as they deal with completely different situations. Rather, they can only reasonably be considered mutually exclusive, successive processes.

In view of all of the above, Appellant respectfully submits that one of skill would not have been motivated to modify *Jamtgaard*'s re-formatting system with the page creation system of *Lipkin*, and therefore that independent claims 1, 11 and 21 are patentable over the applied references.

A(3). Dependent Claims 2-4, 12-14 and 22-24

Appellant again respectfully submits that dependent claims 2-4, 12-14 and 22-24 are allowable at least by virtue of their dependency from independent claims 1, 11 and 21, respectively.

A(4). Dependent Claims 5, 15 and 25

Appellant argued in the August 5, 2005 *Appeal Brief* that the proffered combination of *Jamtgaard*, *Lipkin* and *Maslov* fails to teach or suggest "receiving a request for a script file from a client program; and identifying a document within the document server corresponding to the requested script file," as recited in claims 5 and 25, or "a request reception module configured to

receive a request for a script file from a client program and to identify a document corresponding to the requested script file,” as recited in claim 15.

Specifically, Appellant pointed out that *Maslov* is directed (see FIG. 1) to obtaining fragments 15, 25 of source documents 10, 20 from the internet, and then displaying them in a target window 30, and therefore that *Maslov*’s script is directed to identifying documents from the internet, not documents on the recited document server, as recited in these claims.

The Examiner disagrees, and argues that “the script may be directed towards identifying both documents from the internet and from a specific document server (Figure 6, column 10, lines 60-column 11, 39 of *Maslov*)” (see *Answer*, p. 13, lines 13-15).

Appellant respectfully disagrees. The cited portion of *Maslov* (which is cited for the first time in the Examiner’s *Answer*) indicates that a client 10 may request information from a server 20, which in turn requests the required web page information and performs the source document transformation via a script on the server to prepare a document digest 40. Thus, *Maslov* clearly indicates that a script is provided on a server, which obtains fragments from web pages. There is no teaching or suggestion that the script file is provided at the client 10, or that it is operable to obtain documents on server 20.

Accordingly, Appellant again respectfully submits that dependent claims 5, 15 and 25 are: (1) allowable at least by virtue of their dependency; and (2) separately patentable over the applied references.

A(5). Dependent Claims 6, 16 and 26

Appellant argued in the August 5, 2005 *Appeal Brief* that the proffered combination of *Jamtgaard*, *Lipkin* and *Tadakoro* fails to teach or suggest that “the script file is included within a separate portion of the document,” as recited in claims 6, 16 and 26.

Specifically, Appellant pointed out that *Tadakoro* discloses only script-embedded HTML pages, such as are shown in the instant Application’s prior art Figure 1, and that such script-embedded HTML pages fail to teach or suggest a script file that is in a separate portion of the HTML document.

The Examiner disagrees, and argues that “the script file is embedded into the HTML page, and requires that the parser parses the script and HTML sections of the file separately, thus the sections that contain script file can be thought of as separate because they are processed as existing separately (column 12, lines 45-63 of *Tadokoro*)” (see *Answer*, p. 13, line 21 - p. 14, line 2).

Appellant respectfully disagrees, and submits that the Examiner’s interpretation of “separate” is unreasonable and unsupported. Claims 6, 16 and 26 recite that “the script file is included within a separate portion of the document” (emphasis added). In contrast, *Tadakoro*’s script is clearly included within the HTML portion of the page (*i.e.*, the same portion).

Further, Appellant respectfully submits that it is not relevant to an interpretation of claims 6, 16 and 26 when the script is accessed relative to the HTML, as the Examiner alleges. Rather, claims 6, 16 and 26 specify where on the page the HTML and script are located.

Accordingly, Appellant again respectfully submits that dependent claims 6, 16 and 26 are: (1) allowable at least by virtue of their dependency; and (2) separately patentable over the applied references.

A(6). Dependent Claims 7-9, 17-19, and 27-29

Appellant again respectfully submits that dependent claims 7, 8, 17, 18, 27 and 28 are allowable at least by virtue of their dependency from independent claims 1, 11 and 21, respectively.

A(7). Dependent Claims 10, 20 and 30

Appellant argued in the August 5, 2005 *Appeal Brief* that the proffered combination of *Jamtgaard* and *Lipkin* fails to teach or suggest the replacement of “a first object of the DOM with a different second object,” as recited in claims 10, 20 and 30.

Specifically, Appellant pointed out that: (1) *Jamtgaard* discloses reformatting an HTML document by reformatting various objects (*i.e.*, not replacing any particular DOM objects with other objects); and (2) *Lipkin* is only directed to either insertion of particular command lines, or reformatting existing elements (as discussed above).

The Examiner disagrees, and argues that “as shown in *Jamtgaard*, when the document is converted from HTML to RML, it allows inclusions of different content to different objects, which thus represents a replacement from one object with a second object (column 11, lines 15 - column 12, lines 65 of *Jamtgaard*). These hierarchal differences are evident when comparing the differences between the HTML document and RML document shown in columns 11 and 12, which are merely flattened representations of the DOM tree used to perform the transformation” (see *Answer*, p. 14, lines 9-15).

Appellant respectfully disagrees, and submits that, even if the Examiner's allegation that *Jamtgaard* allows "inclusion of different content to different objects" could be considered correct, such an "inclusion" is not equivalent to the claimed replacement. Rather, the Examiner's cited "inclusion" is the addition of code lines. For example, the objects shown in the sample HTML code in col. 11, lines 18-42 of *Jamtgaard* are also provided in the resulting RML code in col. 12, lines 35-63 of *Jamtgaard*, but with added code. Thus, Appellant respectfully submits that it cannot reasonably be argued that an object has been replaced.

Accordingly, Appellants again respectfully submit that dependent claims 10, 20 and 30 are: (1) allowable at least by virtue of their dependency; and (2) separately patentable over the applied references.

A(8). Dependent Claims 34-36

Appellants again respectfully submit that dependent claims 34-36 are allowable at least by virtue of their dependency from independent claims 1, 11 and 21, respectively.

A(9). Dependent Claims 31-33 and 37-39

Appellant argued in the August 5, 2005 *Appeal Brief* that the proffered combination of *Jamtgaard* and *Lipkin* fails to teach or suggest that "the first object is an HTML file," as recited in dependent claims 31-33, and that "the first object is an HTML file; the transformation instruction is read from a script file located separately from the HTML file; and the HTML file and the script file contain information to indicate their correspondence to each other," as recited in claims 37-39.

Specifically, Appellant pointed out that, since the Examiner has rejected independent claims 1, 11 and 21 as being unpatentable over a combination of *Jamtgaard* in view of *Lipkin*,

the resultant combination must use an XSP model sheet (or its equivalent) as the first object, not the HTML page of *Jamtgaard*.

The Examiner disagrees, and argues that (see *Answer*, p. 14, line 21 - p. 15, line 9):

as shown in both *Jamtgaard* and *Lipkin*, XSL files are used to process the DOM tree with contains objects, the first object of *Jamtgaard* being an HTML page, the objects of *Lipkin* being widgets, which are merely representations of HTML files, thus in both systems the first object may in fact be an HTML file and the cited portion of *Jamtgaard* would not in fact conflict with the teachings of *Lipkin* for this reason (column 11, lines 15- column 12, lines 65 of *Jamtgaard* and column 75, lines 40-50 of *Lipkin*). It is noted that the examiner does not agree that some form of modification to the system of *Jamtgaard* would need to be made to incorporate the teachings of *Lipkin* based on the fact that the *Jamtgaard* teachings were made without the teachings of *Lipkin* in mind. However any necessary modification would not hinder the ability nor the usefulness of the *Jamtgaard* system.

Appellant respectfully disagrees. As the Examiner concedes, *Jamtgaard* fails to teach or suggest any indication of a change in value of a DOM object. Thus, the Examiner alleges that one of skill would modify *Jamtgaard* in view of the system of *Lipkin* to provide a system that allegedly changes the value of a DOM object. In this regard, *Lipkin's* system is specifically based on the use of an XSP model page as its starting point, in order to provide its disclosed functionality (see discussion above). In other words, the entire point of the invention disclosed in *Lipkin* is to use an XSP model page. Thus, any resultant combination of *Jamtgaard* and *Lipkin* would necessarily include the use of such an XSP model page, if one desired to obtain the results of *Lipkin's* functionality.

However, an XSP model page is not an HTML file. Thus, the proffered combination does not teach or suggest the recited features.

Accordingly, Appellants again respectfully submit that dependent claims 31-33 and 37-39 are: (1) allowable at least by virtue of their dependency; and (2) separately patentable over the applied references.

A(10). Dependent Claims 41, 45 and 50

Appellant argued in the August 5, 2005 *Appeal Brief* that the proffered combination fails to teach or suggest: (1) that “the document and the corresponding transformed document are in the same format,” as recited in claims 41, 45 and 50; and (2) that “the same format is HTML,” as recited in claims 41, 45 and 50.

Specifically, Appellant pointed out that: (1) the entire purpose of *Jamtgaard* is to provide a system for transforming an HTML page to a page of another format for display on, for example, a portable device; and (2) the entire purpose of *Lipkin* is to provide a system for transforming an XSP model page (by adding command lines and modifying the format of the XSP model page) into a different format understood by a browser, such as HTML (col. 69, lines 55-65).

The Examiner disagrees, and argues that (see *Answer*, p. 15, line 19 - p. 16, line 5):

in column 54, lines 36-47, Lipkin teaches that pages may be generated and preexisting pages can be customized and/or modified to provide different versions to user agents using the cited embodiment of Lipkin, which is done automatically thus transparent to the user. Lipkin discloses that it would be advantageous to allow different versions of pages to different users, which includes adding and removing information content without changing format (HTML) (column 53, lines 5-19 and column 54, lines 36-47 of Lipkin).

Appellant respectfully disagrees, and respectfully submits that the Examiner’s interpretation of *Lipkin* is incorrect. Actually (as discussed above), *Lipkin* initially provides an

XSP model page, not an HTML page. This is the entire point of *Lipkin* - to format an XSP model accordingly to a needed output. This XSP model page is then transformed (by adding command lines and modifying the format of the XSP model page) into a different format understood by a browser, such as HTML. Thus, *Lipkin*'s documents are, by their very nature, in different formats before and after transformation.

Similarly, it is the entire purpose of *Jamtgaard* to modify the format of an HTML page to another format (*Jamtgaard* Abstract).

Accordingly, Appellants again respectfully submit that dependent claims 41, 45 and 50 are: (1) allowable at least by virtue of their dependency; and (2) separately patentable over the applied references.

A(11). Dependent Claims 42, 47 and 52

Appellant argued in the August 5, 2005 *Appeal Brief* that the proffered combination fails to teach or suggest that "the value is changed in accordance with different users," as recited in claims 42, 47 and 52.

Specifically, Appellant pointed out that, while the cited portion of *Lipkin* indicates that various data might be presented to different users on the final HTML page (*e.g.*, objects A, B and D might be presented to user 1, while objects A, C and D might be presented to user 2), it does not teach or suggest that the values of the objects themselves are any different, or that the values of those objects might be changed, in accordance with a particular user.

The Examiner disagrees, and argues that, "in column 69, lines 1-54, *Lipkin* shows that during a transformation process a DOM tree is accessed and the values of nodes are changed. In this case, the sub-tree, which consists of different object values of HTML content, be [sic]

inserted into the DOM tree is copied to the nodes that require it, thus changing the valued [sic] of those nodes specifically” (see *Answer*, p. 16, lines 12-15).

Appellant respectfully disagrees, and submits that the Examiner did not address Appellant’s arguments directly, as he has not explained how any values are changed in accordance with different users.

Accordingly, Appellants again respectfully submit that dependent claims 42, 47 and 52 are: (1) allowable at least by virtue of their dependency; and (2) separately patentable over the applied references.

A(12). Dependent Claims 43, 48 and 53

Appellant argued in the August 5, 2005 *Appeal Brief* that the proffered combination fails to teach or suggest that “the value is a variable,” as recited in claims 43, 48 and 53, or that “the value of the first object is empty before the first object is transformed,” as recited in claims 44, 49 and 54.

Specifically, Appellant pointed out that: (1) the portion of *Lipkin* cited by the Examiner is directed to the application of an XSLT stylesheet to the XSP model page and that the application of the XSLT stylesheet only adjusts the format of the page, not the underlying data; and (2) that *Lipkin* only discloses adding command lines and modifying the format of the XSP model page, not changing the value of any particular object.

The Examiner disagrees, and argues that “Lipkin also teaches that content may be added to the DOM tree by adding nodes and sub-trees, thus some nodes may exist as empty until data values are assigned to them (column 69, lines 1-54 of *Lipkin*). Lipkin also discloses a method in

which the value of objects in the DOM tree may be variables (column 69, lines 1-24 of *Lipkin*)”
(see *Answer*, p. 16, line 18 - p. 17, line 1).

Appellant respectfully disagrees, and submits that the *Lipkin* does not disclose any variable that has a value that is changed. As shown in the earlier cited examples in cols. 67 and 68 of *Lipkin*, any portion that could be considered to be a variable therein is not changed by *Lipkin*'s system. Rather, such portions are the same in the two examples.

Further, the Examiner's allegation that “content may be added to the DOM tree by adding nodes” is precisely Appellant's point regarding the deficiencies of *Lipkin* relative to the independent claims (as discussed in detail above). As the Examiner recognizes here, the content added to the XSP model page in *Lipkin* is inserted into the DOM, and thus cannot reasonably be read as changing existing objects. Further, since the object does not exist before insertion, it cannot reasonably be argued to be replacing an empty object or variable.

Accordingly, Appellants again respectfully submit that dependent claims 43, 44, 48, 49, 53 and 54 are: (1) allowable at least by virtue of their dependency; and (2) separately patentable over the applied references.

CONCLUSION

In view of the foregoing differences between appealed claims 1-54 and the cited references, Appellants respectfully submit that appealed claims 1-54 are patentable over the applied references.

For the above reasons as well as the reasons set forth in Appeal Brief, Appellants respectfully request that the Board reverse the Examiner's rejections of all claims on Appeal. An early and favorable decision on the merits of this Appeal is respectfully requested.

Respectfully submitted,



Timothy P. Cremen
Registration No. 50,855

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE

23373

CUSTOMER NUMBER

Date: December 19, 2005